



1c639 U.S. PTO
06/26/01

[illegible]

Inventor(s):
Bret O'Rourke
Dawson Dean
Anton Kucer

ATTORNEY'S DOCKET NO. MS1-656US

1 **TECHNICAL FIELD**

2 This disclosure relates to networked client/server systems and to methods
3 of delivering requested content in such systems. More particularly, the subject
4 matter relates to systems and methods for delivering client requested content to the
5 client along with additional content.

6
7 **BACKGROUND**

8 When a client requests a piece of content such as digital video, audio, or
9 some other content from a server, the client typically provides a global address to
10 the content in the form of a Uniform Resource Locator (URL). The server then
11 accesses the addressed content and sends or "streams" it to the client in the form
12 of a continuous digital data stream.

13 There are various file data formats for streaming digital media content and
14 composite media streams. "Advanced Streaming Format" (ASF) is an example of
15 such a data file format. ASF (sometimes referred to as WINDOWS Container
16 Media Format) specifies a way in which multimedia data content is stored,
17 streamed, and presented by a variety of tools, servers, and clients of a number of
18 different multimedia vendors. ASF provides a storage and transmission data file
19 format that encapsulates multimedia data types (e.g., images, audio, and video) as
20 well as embedded text (e.g., a URL) and also allows for synchronizing these
21 objects within a digital data stream. (Further details about ASF are available from
22 Microsoft Corporation of Redmond, Washington).

23 Regardless of which of a number of different streaming file data formats is
24 used, an individual data stream includes a sequence of digital data sets or units.
25 The units represent an image, sound, or some other stimuli that is perceived by a

1 human to be continuously varying. The client renders the units individually, in
2 sequence, to reproduce the original stimuli. For example, an audio data stream
3 comprises a sequence of sample values that are converted to a pitch and volume to
4 produce continuously varying sound. A video data stream comprises a sequence
5 of digitally specified graphics frames that are rendered in sequence to produce a
6 moving picture.

7 In the simplest case, a client requests a single streaming media content file,
8 to reproduce, or “play” a single piece of content such as a song or a video.
9 Alternatively, the client may request a playlist, or “playlist file” that includes a
10 number of different references to individual streaming media content files. Each
11 playlist file includes information such as information to reference specific pieces
12 of content, an order in which to play the referenced content, and other information
13 (e.g., whether to play certain pieces of referenced content more than one time). In
14 other words, a playlist file not only references media content, but also describe
15 how pieces of media content are combined.

16 Playlists do not normally contain the actual media data, but rather particular
17 references (i.e., a URL) to stored media data. As a result, a playlist file is typically
18 small in size, generally only contains text, and is typically easy and
19 computationally inexpensive to modify. (A reference to a single piece of media
20 content may appear in any number of different playlist files).

21 A playlist is typically created in predetermined format. The Synchronized
22 Multimedia Integration Language (version. 2.0), referred to as “SMIL” is an
23 example of such a predetermined format. SMIL is an extension of the World
24 Wide Web Consortium (W3C) standard Extensible Markup Language (XML).
25 SMIL provides syntax and structure to define both high-level instructions and data

1 corresponding to the content referenced by a playlist. (The specification for SMIL
2 is well understood in the computing industry).

3 A playlist can be implemented on a client computer or on a server
4 computer. When a client implements a playlist, the playlist is typically
5 downloaded from a server. The client interprets the downloaded playlist file to
6 present a series of requests to the server, for every piece of content represented in
7 the playlist. A server, upon receipt of a media content request from a client, is
8 generally not aware that the client is requesting media content that is referenced in
9 a client implemented, or "client-side" playlist file. This is because use of a client-
10 side playlist is indistinguishable from a client requesting a server to play one or
11 more respective pieces of media content one-after-the-other.

12 A server implemented playlist, or "server-side" playlist is maintained by a
13 server and is not downloaded to a client. To access media content represented in a
14 server-side playlist, a client typically selects a URL that identifies both a server
15 and a particular playlist file. Responsive to receiving a request from the selection
16 of such a URL, the identified server accesses the requested playlist and streams, or
17 "plays" the media content referenced by the playlist to the requesting client, one
18 piece of media content at a time.

19 Regardless of whether a playlist is implemented on a server or on a client, a
20 playlist can play a substantially important role in building a business based on
21 advertising, source branding, service branding, and/or other revenue basis. For
22 example, a playlist file can allow a content provider (e.g., an Internet radio station)
23 to embed, or combine an advertisement, a brand name, and/or other content such
24 as multimedia content previews, radio-station identifiers and/or the like with
25 scheduled media content. Every time that a content provider chooses to modify an

1 advertisement, branding information, and/the like, that is embodied in a playlist,
2 the content provider must typically modify the playlist to excise the old content to
3 incorporate the new content. Or, the content provider can remove the old playlist
4 and create a completely new playlist to reflect the change(s). Such playlist
5 modifications and may be brought about for any number of reasons such as
6 changes in business methods, regulatory constraints, audience demographics,
7 and/or the like. It can be appreciated that a content provider such as a radio or
8 television station may have any number of playlists (e.g., one playlist, one-
9 hundred playlists, one thousand playlists, or more) that need to be modified, and or
10 created to reflect such changes. Thus such playlist media content changes
11 typically require a substantial amount of administrative overhead.

12 13 SUMMARY

14 Various embodiments for automatically associating new or different media
15 content with a particular playlist are described. In one implementation, a server
16 receives a request from a client for one or more content items. In response, the
17 server identifies a data structure that refers to one or more further content items.
18 Next, the server modifies the data structure to reference the content item or items
19 requested by the client. Finally, the server provides the content items that are
20 referenced by the modified data structure to the client.

21 The server can insert the reference to the client-requested content item into
22 a particular point in a sequence of references in the data structure. The data
23 structure can be created by modifying a base data structure or by copying a base
24 data structure.

1 The client requested content item can be a streaming media content item,
2 and the data structure can be a playlist that references a number of other content
3 items such as streaming media items. Alternatively, the client-requested content
4 item can be a Web page specification document, which is to be inserted in a data
5 structure such as another Web page specification.

6 7 **BRIEF DESCRIPTION OF THE DRAWINGS**

8 Fig. 1 is a block diagram that illustrates aspects of an exemplary system to
9 deliver additional content along with requested content from a server to a client.

10 Fig. 2 is a block diagram that illustrates aspects of an exemplary playlist
11 wrapper data structure that refers to one or more content items.

12 Fig. 3 is a block diagram that illustrates aspects of an exemplary playlist
13 wrapper data structure or playlist that refers to one or more content items.

14 Fig. 4 is a block diagram that illustrates aspects of an exemplary copy of a
15 playlist wrapper data structure, or playlist that refers to one or more content items.

16 Fig. 5 is a block diagram that illustrates aspects of a modified playlist
17 wrapper data structure or playlist that refers to one or more content items along
18 with a client requested content item.

19 Fig. 6 is a flowchart that illustrates aspects of an exemplary methodology
20 for a server to stream content items to a client.

21 Fig. 7 is a block diagram that illustrates aspects of an exemplary server
22 computer to deliver additional content along with requested content to a client.

1 **DETAILED DESCRIPTION**

2 The following description sets forth a specific embodiment of a server
3 component and methodology that incorporates elements recited in the appended
4 claims. The embodiment is described with specificity in order to meet statutory
5 requirements. However, the description itself is not intended to limit the scope of
6 this patent. Rather, the inventors have contemplated that the claimed subject
7 matter might also be embodied in other ways, to include different elements or
8 combinations of elements similar to the ones described in this document, in
9 conjunction with other present or future technologies.

10
11 **Exemplary System Environment**

12 Fig. 1 shows a client/server network system 100 and environment.
13 Generally, the system includes one or more network server computers 102 and
14 multiple network client computers 104. The computers communicate with each
15 other over a data communications network 106 such as the Internet. The data
16 communications network might also comprise intranets, local-area networks and
17 private wide-area networks. The system has associated data storage media 106.

18 In one embodiment, server 102 is a streaming media content server that
19 receives URLs or other references that indicate streaming media content, files,
20 and/or sources. For example, the URL might identify an ASF (.asf) file. A variety
21 of requests from the client are typically received by server 102, and might identify
22 requested content in a variety of different forms other than the URL form.
23 Furthermore, some requests might not be processed in accordance with the
24 technique described below. However, at least one subset of the received requests
25

1 includes requests to one or more different content items that will be processed in
2 accordance with the described technique.

3 In response to receiving the content request, the server 102 identifies or
4 creates a data structure, or content template that refers to one or more other
5 content items. In the described embodiment, the content template comprises a
6 playlist. The playlist or other type of data structure can be created in a number of
7 different data formats. SMIL provides an example of a suitable format.

8 The data structure acts as a wrapper or content template because the data
9 structure specifies content that will wrap or frame at least a subset of the content
10 sent by a server 102 to a client 104.

11 After identifying the appropriate base data structure, the server 102
12 modifies the data structure to reference the requested content item by inserting the
13 requested URL into the playlist's sequence of URLs. In most cases, this
14 modification will be performed on a copy of the base data structure. After the
15 base playlist has been modified in this manner, server 102 provides the content
16 items referenced by the modified data structure or playlist to the client 104.

17 Fig. 2 shows an exemplary data structure 200 to reference one or more
18 items of content. The data structure includes a sequence of one or more references
19 202 to content items. In the described implementation, the references are URLs.
20 Additionally, the data structure includes a placeholder 204, or flag that identifies a
21 particular point in a sequence where the reference to the client requested content is
22 to be placed. The placeholder 204 can be located at any position with respect to
23 the references 202.

24 There can be any number of placeholders 204 in the data structure 200,
25 each placeholder identifying a particular point in a sequence where the reference

1 to the client-requested content is to be placed. If there is more than one
2 placeholder, the total number of placeholders is analogous to a repeat count of
3 how many times the client requested content will be referenced in the data structure.
4 Optionally, the number and position of each placeholder is indicated in the client's
5 request for the content item.

6 Although the implementation that is described above has been applied to
7 playlists and streaming media, there are other types of content that might benefit
8 from this approach. For example, the client's request might be for any other type
9 of a document or file such as a Web page (e.g., an HTML document). In this case,
10 the data structure 200 can be represented in the file/document/Web page
11 specification with a placeholder indicating where the requested content, or
12 reference to the content is to be inserted.

13 It can be appreciated that other types and combinations of content besides
14 streaming media files are also possible such as bitmaps, JPEGs, MPEGs, and/or
15 the like.

16 Fig. 3 is a block diagram that shows an exemplary base playlist 206 that
17 references one or more items of streaming media content. The playlist is one
18 example of a data structure 200 of Fig. 2. In this implementation, the playlist
19 references streaming media content items by identifying the content items with
20 respective URLs 208. The playlist also includes a placeholder 210, or flag to
21 identify a particular point in a sequence where a reference to client requested
22 content is placed. The placeholder can be located at any position with respect to
23 the one or more references or URLs.

24 Fig. 4 is a block diagram that shows an exemplary copy 212 of a base data
25 structure 206 or playlist. The copy is modified to reference client requested

content item(s). Because playlist 212 is a copy of playlist 206, playlist 212 includes streaming media content references 208 and placeholder 210.

A client plays a temporary copy 212 of the wrapper playlist file 206, so modifying the temporary copy for one client request will not affect any other concurrent or future client requests. Thus, a server is able to apply a same wrapper playlist to any number of different clients that request any number of different original URL's. This is because each requesting client will play their own requested URL in a private copy of the wrapper playlist.

A URL 214 represents an item of content requested by a client. In this example, the client's content request refers to streaming media content file "newstuff.asf". Server 102 modifies playlist 212 to reference the requested content item. This modification is illustrated in Fig. 5.

Fig. 5 is a block diagram that shows an exemplary playlist 212 after it has been modified. Specifically, the server 102 has inserted a URL 214 into the playlist 212 at the location identified by a placeholder 210. A server can use this technique to "wrap" other content items around a client-requested content item. The client 104 does is not required to do anything special to receive the other content items. This is because the server automatically translates the client's request for a particular content item into a request for the particular content item and the other content items.

To illustrate this, consider a playlist that includes the HTML instruction sequence shown in Table 1.

Table 1
EXAMPLE OF A PLACEHOLDER IN A WRAPPER PLAYLIST

```

<HTML>
<SEQ>
  <MEDIA src="http://advertisements/ advertisement1.asf ">
  <MEDIA src="%url%"/>
  <MEDIA src=" http://advertisements/advertisement2.asf ">
</SEQ>
</HTML>

```

A placeholder 214 in the above HTML sequence is identified by "%url%". The server replaces every occurrence of the string "%url%" with a URL requested by the client. For example, consider a playlist that includes the HTML sequence shown in Table 2.

Table 2
EXAMPLE OF WRAPPING OTHER URLS AROUND A
A CLIENT REQUESTED URL

```

<HTML>
<SEQ>
  <MEDIA src="http://advertisements/ advertisement1.asf"/>
  <MEDIA
src="http://serverName/FeatureMovies/BigActionMovie.asf"/>
  <MEDIA src="http://advertisements/advertisement2.asf ">
</SEQ>
</HTML>

```

Note that the "%url%" placeholder has been replaced with a client requested URL 214 (<http://serverName/FeatureMovies/BigActionMovie.asf>).

In these examples, the playlists of Tables 1 and 2 are illustrated using the SMIL data format. However, the SMIL format is used for illustration purposes

only, and a wrapper playlist can be generated and/or modified using instructions and/or data represented in one of a number of other multimedia integration formats.

Using a playlist structure and technique as described, allows a server to add advertisements, previews of coming attractions, or other content to a playlist sequence without having to modify an original playlist, or change a URL that corresponds to the original playlist. A data structure or playlist is not restricted to placing advertisements before a requested URL. Such playlists can be used to specify that a client will receive any stream of content that can be described by the playlist language, and that the original requested URL may appear anywhere and any number of times in this sequence of content. Moreover, as discussed above, any other playlist attributes that can be applied to content, such as repeat count and/or the like, can also be applied to the original requested content reference (e.g., a URL).

Furthermore, a data structure such as this allows an administrator to specify an advertising policy by editing the content of the playlist. The policy can easily be changed, by simply editing the base playlist. As a result, the discussed problem of configuring an advertising policy is turned into a much simpler problem of authoring an advertising template or playlist to frame or wrap all content. Authoring content is a well-understood problem and many existing tools for authoring Web content can help with this task.

In some cases, a server may store a standard playlist that is applied to all incoming client requests for content items. In other implementations, a collection of base playlists can be stored for use with different types of requests. Requests can be distinguished, for example, by a storage location of the referenced content,

1 by a file name or extension indicated in the request, and/or by a variety of other
2 distinguishing factors.

3 As discussed above, base data structures or wrappers can be useful for other
4 types of servers besides streaming media servers. For example, a Web server
5 typically serves Web pages that contain multiple references or URLs specified in a
6 format such as HTML (hypertext markup language). A document such as this can
7 be used as a template or wrapper according to the novel procedures discussed
8 above, by including a placeholder indicating where embedded content should be
9 located. When a client requests a Web page specification by supplying a URL
10 reference, the server inserts this URL reference into the wrapper, in place of the
11 placeholder, and then serves the resulting Web page.

12 To illustrate this, consider that a client requests a URL, for example,
13 "http://server/doc1.htm". The Web server responds by making a temporary copy
14 of a "wrapper" HTML document. The Web server then modifies the temporary
15 copy by inserting the originally requested URL ("http://server/doc1.htm") into the
16 contents of the wrapper document.

17 The "wrapper" HTML document may place banner advertisements and
18 other helpful links around the originally requested content. When the modified
19 temporary copy of the wrapper html document is presented to the user, the
20 contents of the original URL "http://server/doc1.htm" might then appear in the
21 middle of the wrapper document.

22 Exemplary Procedure

23
24 Fig. 6 is a flowchart diagram that shows an exemplary procedure 600 to
25 stream data from a server to a client. At block 610, the procedure receives a

1 request from a client for one or more content items. At block 612, the procedure
2 identifies a data structure that refers to one or more further content items. At
3 block 614, the procedure modifies the data structure to reference the requested
4 content items. At block 616, the procedure provides the requested content items
5 (block 610) referenced by the modified data structure (block 614) to the requesting
6 client (block 610).

7 8 **Exemplary Computer Environment**

9 Fig. 7 shows components of computer 700 that forms a suitable
10 environment for a server computer 102 of Fig. 1. The components shown in Fig. 7
11 are only examples, and are not intended to suggest any limitation as to the scope of
12 the functionality of the subject matter; the subject matter is not necessarily
13 dependent on the features shown in Fig. 7.

14 Generally, various different general purpose or special purpose computing
15 system configurations can be used. Examples of well known computing systems,
16 environments, and/or configurations that may be suitable for use with the subject
17 matter include, but are not limited to, personal computers, server computers, hand-
18 held or laptop devices, multiprocessor systems, microprocessor-based systems, set
19 top boxes, programmable consumer electronics, network PCs, minicomputers,
20 mainframe computers, distributed computing environments that include any of the
21 above systems or devices, and the like.

22 The functionality of a computer is embodied in many cases by computer-
23 executable instructions, such as program modules that are executed by the
24 computer. Generally, a program module includes routines, programs, objects,
25 components, data structures, etc. that perform particular tasks or implement

1 particular abstract data types. Tasks might also be performed by remote
2 processing devices that are linked through a communications network. In a
3 distributed computing environment, program modules may be located in both local
4 and remote computer storage media.

5 The instructions and/or program modules are stored at different times in the
6 various computer-readable media that are either part of the computer or that can be
7 read by the computer. Programs are typically distributed, for example, on floppy
8 disks, CD-ROMs, DVD, or some form of communication media such as a
9 modulated signal. From there, they are installed or loaded into the secondary
10 memory of a computer. At execution, they are loaded at least partially into the
11 computer's primary electronic memory. The subject matter described herein
12 includes these and other various types of computer-readable media when such
13 media contain instructions programs, and/or modules for implementing the steps
14 described below in conjunction with a microprocessor or other data processors.
15 The subject matter also includes the computer itself when programmed according
16 to the methods and techniques described below.

17 For purposes of illustration, programs and other executable program
18 components such as the operating system are illustrated herein as discrete blocks,
19 although it is recognized that such programs and components reside at various
20 times in different storage components of the computer, and are executed by the
21 data processor(s) of the computer.

22 With reference to Fig. 7, the components of computer 700 may include, but
23 are not limited to, a processing unit 720, a system memory 730, and a system bus
24 721 that couples various system components including the system memory to the
25 processing unit 720. The system bus 721 may be any of several types of bus

1 structures including a memory bus or memory controller, a peripheral bus, and a
2 local bus using any of a variety of bus architectures. By way of example, and not
3 limitation, such architectures include Industry Standard Architecture (ISA) bus,
4 Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video
5 Electronics Standards Association (VESA) local bus, and Peripheral Component
6 Interconnect (PCI) bus also known as the Mezzanine bus.

7 Computer 700 typically includes a variety of computer-readable media.
8 Computer-readable media can be any available media that can be accessed by
9 computer 700 and includes both volatile and nonvolatile media, removable and
10 non-removable media. By way of example, and not limitation, computer-readable
11 media may comprise computer storage media and communication media.
12 Computer storage media includes volatile and nonvolatile, removable and non-
13 removable media implemented in any method or technology for storage of
14 information such as computer-readable instructions, data structures, program
15 modules, or other data. Computer storage media includes, but is not limited to,
16 RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM,
17 digital versatile disks (DVD) or other optical disk storage, magnetic cassettes,
18 magnetic tape, magnetic disk storage or other magnetic storage devices, or any
19 other medium which can be used to store the desired information and which can be
20 accessed by computer 710.

21 Communication media typically embodies computer-readable instructions,
22 data structures, program modules or other data in a modulated data signal such as
23 a carrier wave or other transport mechanism and includes any information delivery
24 media. The term "modulated data signal" means a signal that has one or more of
25 its characteristics set or changed in such a manner as to encode information in the

1 volatile/nonvolatile computer storage media that can be used in the exemplary
2 operating environment include, but are not limited to, magnetic tape cassettes,
3 flash memory cards, digital versatile disks, digital video tape, solid state RAM,
4 solid state ROM, and the like. The hard disk drive 741 is typically connected to
5 the system bus 721 through an non-removable memory interface such as interface
6 740, and magnetic disk drive 751 and optical disk drive 755 are typically
7 connected to the system bus 721 by a removable memory interface such as
8 interface 750.

9 The drives and their associated computer storage media discussed above
10 and illustrated in Fig. 7 provide storage of computer-readable instructions, data
11 structures, program modules, and other data for computer 700. In Fig. 7, for
12 example, hard disk drive 741 is illustrated as storing operating system 744,
13 application programs 745, other program modules 746, and program data 747.
14 Note that these components can either be the same as or different from operating
15 system 734, application programs 735, other program modules 736, and program
16 data 737. Operating system 744, application programs 745, other program
17 modules 746, and program data 747 are given different numbers here to illustrate
18 that, at a minimum, they are different copies. A user may enter commands and
19 information into the computer 700 through input devices such as a keyboard 762
20 and pointing device 761, commonly referred to as a mouse, trackball, or touch
21 pad. Other input devices (not shown) may include a microphone, joystick, game
22 pad, satellite dish, scanner, or the like. These and other input devices are often
23 connected to the processing unit 720 through a user input interface 760 that is
24 coupled to the system bus, but may be connected by other interface and bus
25 structures, such as a parallel port, game port, or a universal serial bus (USB). A

1 monitor 791 or other type of display device is also connected to the system bus
2 721 via an interface, such as a video interface 790. In addition to the monitor,
3 computers may also include other peripheral output devices such as speakers 797
4 and printer 796, which may be connected through an output peripheral interface
5 795.

6 The computer may operate in a networked environment using logical
7 connections to one or more remote computers, such as a remote computer 780.
8 The remote computer 780 may be a personal computer, a server, a router, a
9 network PC, a peer device or other common network node, and typically includes
10 many or all of the elements described above relative to computer 700, although
11 only a memory storage device 781 has been illustrated in Fig. 7. The logical
12 connections depicted in Fig. 7 include a local area network (LAN) 771 and a wide
13 area network (WAN) 773, but may also include other networks. Such networking
14 environments are commonplace in offices, enterprise-wide computer networks,
15 intranets, and the Internet.

16 When used in a LAN networking environment, the computer 700 is
17 connected to the LAN 771 through a network interface or adapter 770. When used
18 in a WAN networking environment, the computer 700 typically includes a modem
19 772 or other means for establishing communications over the WAN 773, such as
20 the Internet. The modem 772, which may be internal or external, may be
21 connected to the system bus 721 via the user input interface 760, or other
22 appropriate mechanism. In a networked environment, program modules depicted
23 relative to the computer 700, or portions thereof, may be stored in the remote
24 memory storage device. By way of example, and not limitation, Fig. 7 illustrates
25 remote application programs 785 as residing on memory device 781. It will be

1 appreciated that the network connections shown are exemplary and other means of
2 establishing a communications link between the computers may be used.

3 4 Conclusion

5 Although various implementations of the described subject matter have
6 been described in language specific to structural features and/or methodological
7 operations, it is to be understood that the subject matter defined in the appended
8 claims is not necessarily limited to the specific features or steps described. Rather,
9 the specific features and operations are disclosed as exemplary forms of
10 implementing the claimed subject matter.

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25